Instituto Tecnológico de Costa Rica

Escuela de Ingeniería Electrónica

Manual de Usuario

Nodo de recolección y procesamiento de datos basado en GNU/Linux

Realizado por:

Ing. Carmen Chan Zheng

Julio, 2012

El presente manual es una guía para el usuario que contiene las instrucciones para la instalación y el uso del nodo de recolección, envío y procesamiento de datos del proyecto *Predicción Remota de Fallas Estructurales en Puentes* del grupo de investigación *eBridge* del Instituto Tecnológico de Costa Rica.

El proyecto consiste en una red de sensores colocados en varios puntos estratégicos de un puente, cada sensor se encuentra acoplado a un dispositivo denominado nodo, el cual recolecta los datos y los envía hacia un nodo principal (nodo de recolección, envío y procesamiento de datos) ubicado a una distancia máxima de 100 metros del puente.

Figura 1 Nodo de recolección, envío y procesamiento de datos

Los módulos o componentes que forman el nodo son:

- Adaptador de 5V
- BeagleBoard xM
- Tarjeta microSD
- Datacard
- Gateway

INSTALACIÓN DEL SISTEMA OPERATIVO

Para la configuración de un sistema operativo en la BeagleBoard se requiere el uso de una memoria microSD. La siguiente guía se encuentra basada en el método mencionado en la página *elinux.org*[1].

1. Se descarga una imagen de la distribución Ubuntu de Linux. Esta imagen contiene los requerimientos mínimos que se necesitan para inicializar cualquier sistema. El enlace que contiene las imágenes corresponde a:

http://rcn-ee.net/deb/rootfs/.

En este sitio web se encuentra distintas versiones de la distribución: *oneiric, precise, natty, maverick*, entre otras. La versión implementada en el proyecto es la versión 11.04 (*natty*):

http://rcn-ee.net/deb/rootfs/natty/ubuntu-11.04-r7-minimal-armel.tar.xz

2. Al finalizar la descarga se procede a descomprimir el archivo. En la ventana de consola se ejecuta el siguiente comando:

sudo tar xJf ubuntu-11.04-r7-minimal-armel.tar.xz

Posteriormente, entra al directorio de la carpeta recién descomprimida:

cd ubuntu-11.04-r7-minimal-armel

3. Se busca el directorio de la tarjeta microSD mediante el siguiente comando:

sudo ./setup_sdcard.sh --probe-mmc

[1] http://elinux.org/BeagleBoardUbuntu

En la ventana de consola se despliega el siguiente mensaje:

```
Are you sure? I Don't see [/dev/idontknow], here is what I do see...

fdisk -l:

Disk /dev/sda: 500.1 GB, 500107862016 bytes <- x86 Root Drive

Disk /dev/mmcblk0: 3957 MB, 3957325824 bytes <- MMC/SD card

mount:

/dev/sda1 on / type ext4 (rw,errors=remount-ro,commit=0) <- x86 Root Partition
```

El directorio señalado por MMC/SD card corresponde al directorio de la tarjeta microSD, en este caso /dev/mmcblk0.

4. En la carpeta se encuentra un script que instala todos los archivos de la imagen en la tarjeta microSD:

sudo ./setup_sdcard.sh --mmc *directoriomicroSD* --uboot beagle_xm

5. Al finalizar la ejecución del script, se expulsa la tarjeta microSD, y un par de segundos después se reinserta la tarjeta en la computadora. En la tarjeta se crearon dos particiones que se encuentran montada en los siguientes directorios:

/media/rootfs

/media/boot

CONFIGURACIÓN INICIAL DEL S.O

Al finalizar la preparación de la tarjeta microSD, se procede a copiar los archivos de configuración del proyecto a la microSD. En una ventana de consola se procede a crear una carpeta dentro de la tarjeta:

sudo mkdir /media/rootfs/Nodo

Seguidamente se copia todos los archivos de configuración a dicha carpeta mediante el comando *cp*. Los archivos de configuración se encuentran disponibles en el siguiente enlace: http://www.ie.itcr.ac.cr/jcarvajal/ebridge/Nodo

Al finalizar la creación de los archivos del proyecto se desmonta las particiones:

sudo umount /media/rootfs

sudo umount /media/boot

Luego, se inserta dicha tarjeta en la ranura de la BeagleBoard. Sin embargo, para visualizar la consola recién instalada se debe utilizar un cable de pruebas Serial-USB para la comunicación entre la tarjeta y la computadora con Linux para el monitoreo de la primera. En una ventana de consola de la computadora se ejecuta la siguiente orden:

sudo screen /dev/ttyUSB* 115200

*:0-9 el que se encuentre acoplado el cable serial

Con la orden anterior, se visualiza la consola instalada en la *BeagleBoard.*

Al terminar la inicialización de la BeagleBoard se procede a insertar el nombre de usuario y la clave correspondiente. El nombre de usuario corresponde a *ubuntu* y la clave *temppwd.*

Al inicializar por primera vez el Ubuntu, el usuario debe configurar algunos archivos para realizar las siguientes tareas:

- Automatización de los procesos
- Conexión de los módulos USB al directorio /dev/
- Ajuste zona horaria
- Ajuste de la fecha y hora al inicio del sistema

Las tareas mencionadas anteriormente se integraron en un script ejecutable para evitar la configuración manual. Todos los archivos de configuración se colocan sobre un mismo directorio.

La configuración inicial se requiere de Internet, por lo que se requiere un cable Ethernet. En la consola de la tarjeta se debe introducir el comando para habilitar la conexión:

sudo dhclient eth0

Con la conexión establecida, se entra al directorio con los archivos del sistema:

cd /Nodo

Se le da los permisos de ejecución al script:

sudo chmod 777 installeBridge.sh

Por último, se ejecuta el script de instalación:

./installeBridge.sh

ARCHIVOS CONFIGURABLES POR EL USUARIO

En esta sección se comenta los archivos que pueden ser modificadas por el usuario.

75-tty-description.rules

Este archivo se encarga de acoplar los módulos *Gateway y Datacard* al directorio */dev/gateway y /dev/datacard* respectivamente.

Sin embargo, no todos los Datacard tienen el mismo *id vendor* ni el *id product*. Para comprobar el *id* de los dispositivos se ejecuta en la consola el comando *lsusb.*

Por tanto, en el archivo *75-tty-description.rules* se modifica el *id vendor y id product* de la siguiente línea:

```
# do not edit this file, it will be overwritten on update

……

SUBSYSTEMS=="usb", DRIVERS=="usb",
⇒ ATTRS{idVendor}=="id vendor", ATTRS{idProduct}=="id
product", ENV{ID_VENDOR_ID}="$attr{idVendor}",
ENV{ID_MODEL_ID}="$attr{idProduct}", SYMLINK+="datacard"

……
```

Fusion.sh

En el presente sistema se encuentra implementado dos operadora de telefonía celular: *kolbi* y *Movistar.* En este archivo se modifica cual operadora será utilizada en la aplicación. El bloque de código por modificar:

```
elif ls NumAck; then
            ./Recoleccion.py
            ./Formato.py
⇒           pon ICE &
⇒ #         pon Movistar &
            sleep 15
            ./EnvioSFTP.py
            poff
            sleep 10
……
```

Se comenta la operadora que **NO** se usará.

Recolección.py

Este script se modifica en caso de que se agregue un nodo más: *nodoX*. Se modifican los siguientes bloques de código:

Bloque 1:

```
#**********************Variables de ayudas***********
      y=0
      datos=0
      i = 0
      nodoA=0
      nodoB=0
#*************Agregar nueva variable de nuevo nodo aca******
      nodoX=0
```

Bloque 2:

```
      if nodoA == n and nodoB == n:    ##Agregar nuevo nodo nodoX==n
                  print "Listo"
                  break
```

Bloque 3:

```
#************************Nuevo Nodo*******************************
elif nodo == "NodoX:" and nodoX<n:     ##Se tiene que recoger cantidad
                                        de muestras en LVDT
      time1=''
      print "NodoX"
      serie = ser.readline().strip()
      datos= serie
      print datos
      time1 = time.strftime('%T') # FORMATO HORA:MINUTOS:SEGUNDOS
      obj=csv.writer(f,dialect='excel',delimiter=',',quoting=csv.QUOTE
      _MINIMAL)
      obj.writerow([time1,'B',datos])##Guarda   hora   actual   y   el
                                     promedio   obtenido   de   las
                                     variables
      nodoX=nodoX+1
```

Además, en la siguiente función:

Recoleccion=TimeoutFunction(Recoleccion,960)

Esta función limita la ejecución del script a 960 segundos, en caso de que algún nodo se queda sin batería, después de ese tiempo se interrumpe el código. Se puede modificar ese límite de tiempo de acuerdo a las necesidades del usuario.

Nuevas operadora de telefonía celular

Si el usuario emplea una nueva operadora de telefonía celular se debe crear dos archivos. El primer archivo se crea en el directorio ***/etc/ppp/peers/nombre_operadora*** con el siguiente código***:***

```
/dev/datacard
115200
noauth
usepeerdns
modem -detach
crtscts
asyncmap 0
defaultroute
noipdefault
lock
connect "/usr/sbin/chat    -v -f /etc/ppp/chat_nombreoperadora"
```

El segundo archivo se crea con el nombre ***chat_nombreoperadora*** en el directorio ***/etc/ppp/:***

```
ABORT BUSY
ABORT 'NO CARRIER'
ABORT VOICE
ABORT 'NO DIALTONE'
ABORT 'NO DIAL TONE'
ABORT 'NO ANSWER'
ABORT DELAYED
""    'ATZ'
OK    'AT&F'
OK    'ATQ0 V1 E1'
OK    'AT&D2 &C1'
OK    'AT+FCLASS=0'
OK    'ATS0=0'

OK    AT+CGDCONT=1,"IP","APN OPERADORA"
OK    AT+CGATT=1
OK    ATD*99***1#
CONNECT '\c'
```

En este archivo se coloca el APN de la operadora seleccionada.

Posteriormente, en el archivo Fusion.sh se comenta las líneas de las operadoras y se coloca una nueva línea:

pon nombre_operadora &

Nota: Para más información del diseño de las rutinas de cada script consultar la Tesis titulada “Nodo de recolección y procesamiento de datos basado en GNU/Linux”. Disponible en el repositorio de la Biblioteca Biblioteca José Figueres Ferrer del Instituto Tecnológico de Costa Rica.